

5. ☐ The entire disclosure of the prior application referred to above is considered to be part of the accompanying application and is hereby incorporated by reference herein.
6. ☐ Microfiche Computer Program (Appendix)
7. ☐ Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
 - ☐ Computer Readable copy
 - ☐ Paper Copy (identical to computer copy)
 - ☐ Statement verifying identity of above copies
8. ☒ Assignment Papers
9. Power of Attorney
 - ☒ Is attached.
 - ☐ The power of attorney appears in the original papers of the prior application.
 - ☐ Since the power does not appear in the original papers, a copy of the power in the prior application is enclosed.
10. ☐ Information Disclosure Statement (IDS)
 - ☐ Copies of IDS Citations
11. Amendments
 - ☐ A preliminary amendment is enclosed.
 - ☐ Cancel in this application claim(s) _____ before calculating the filing fee. At least one independent claim is retained for filing purposes.
 - ☐ Amend the specification by inserting before the first line the sentence: _____.
12. ☒ Return Receipt Postcard
13. Small Entity Status
 - ☐ A small entity statement is enclosed.
 - ☐ A small entity statement was filed in the prior non-provisional application and such status is still proper and desired.
 - ☐ Is no longer claimed.
14. ☐ Priority of foreign application number _____, filed on _____ in _____ is claimed under 35 U.S.C. §§ 119(a)-(d)
15. ☐ Petition under 37 C.F.R. § 136 for Extension of Time
16. ☐ Other: _____

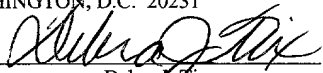
Address all future correspondence to:

Robert C. Kowert
 Conley, Rose, & Tayon, P.C.
 P.O. Box 398
 Austin, Texas 78767
 Phone: (512) 476-1400 Fax: (512) 703-1250

Signature _____
 Name Robert C. Kowert
 Registration No. 39,255
 Date April 21, 2000

PATENT
5181-36000
P4233

"EXPRESS MAIL" MAILING LABEL
NUMBER EL591687622US
DATE OF DEPOSIT APRIL 21, 2000
I HEREBY CERTIFY THAT THIS PAPER OR
FEE IS BEING DEPOSITED WITH THE
UNITED STATES POSTAL SERVICE
"EXPRESS MAIL POST OFFICE TO
ADDRESSEE" SERVICE UNDER 37 C.F.R. §
1.10 ON THE DATE INDICATED ABOVE
AND IS ADDRESSED TO THE ASSISTANT
COMMISSIONER FOR PATENTS,
WASHINGTON, D.C. 20231


Debra J. Tix

**MECHANISM FOR EFFICIENT SCHEDULING
OF COMMUNICATION FLOWS**

by

Kenneth A. Ward

Robert C. Kowert
Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, Texas 78767-0398
Phone: (512) 476-1400

BACKGROUND OF THE INVENTION

1. Field of the Invention

5 This invention relates to servicing queues of communication requests in computing systems. For example, system area network systems may support hundreds or thousands of I/O channels which share a network fabric or other communication link. This invention relates to how the bandwidth of the fabric or link may be shared amongst the communication channels.

2. Description of the Related Art

10 Computer systems I/O performance has become critical in today's applications. Applications such as web servers, internet based e-commerce, online transaction processing, mail and messaging, and other internet, intranet and extranet applications are extremely I/O intensive. A computer's processor may need to frequently communicate with storage systems, network cards, and even other processors over limited I/O connections. The I/O portion of computer systems is rapidly becoming a bottleneck on performance. Recently, new I/O architectures have been proposed to improve upon the I/O bottleneck problem. Some of these proposals include system area network solutions employing numerous I/O channels sharing a switched interconnect fabric. Some recent I/O architecture proposals include Future I/O and NGIO which have merged into System I/O which is now controlled by the InfiniBand Trade Association.

25 A common theme in modern I/O architectures is the need to service many communication requests to many different communication channels that are sharing a limited total bandwidth. One problem is how to divide or allocate the available bandwidth amongst the various communication requests. Turning now to Figure 1, a computing system is shown in which numerous communication channels are supported. A typical computing system may include multiple CPU's 102 coupled to a host bus 103.

A system memory 104 and memory controller 106 may also be coupled to host bus 103 and shared by CPU's 102. Alternatively, or additionally, each CPU 102 may have its own local memory and memory controller. A host channel adapter 108 couples the host bus elements to various I/O resources. Host channel adapter 108 may be integrated in a single component with memory controller 106. A switch 110 may be employed to select amongst various I/O channels to connect to I/O devices 112. Switch 110 may include multiple switches arranged in parallel or in a hierarchical manner. Each I/O device or subsystem 112 may include a target channel adapter for interfacing to the I/O fabric 114.

The host channel adapter 108 serves to decouple CPU's 102 from I/O communications. Various applications executing on CPU's 102 may make I/O or communication requests to host channel adapter 108 for various I/O resources. Host channel adapter 108 services communication requests and returns the results to the requestor. Requests to the same channel may be placed in a queue awaiting service. Host channel adapter 108 services the communication requests within the constraints of the limited bandwidth available through switch 110 and I/O fabric 114. Figure 1 is merely an illustration of one type of system employing multiple communication channels sharing a limited bandwidth. The problem of providing service for numerous communication channels can be found in many different architectures and computing applications.

Turning now to Figure 2, a more conceptual illustration of queuing communication requests for different communication channels is provided. A system may be configured to provide for up to a maximum number of communication request queues 122. For example, a system may support up to 64K queues. Each queue establishes a sequence of operations to be performed on a particular communication channel. At any given time each queue may or may not have operations to execute. The servicing of the communication queues 122 may be performed by a host or local channel adapter 124. The local channel adapter 124 decides which request queue 122 is serviced over its communication channel through the communication fabric 126 to the appropriate remote channel adapter and communication device 128. Since the bandwidth of the

communication fabric 126 is limited, the local channel adapter 124 must decide how best to distribute the limited bandwidth amongst the communication request queues 122. It is typically desirable that all queues be serviced in a fair fashion. Fairness may mean equal access to a given amount of bandwidth based upon a particular service class. Different
5 service classes may be allocated different amounts of total bandwidth.

When a number of communication channels share bandwidth of physical media, as illustrated in Figure 2, each channel may be viewed as a queue with traffic/packets to be sent over the media or fabric. Typically the number of channels greatly exceeds the
10 number of physical media paths that are shared by the physical devices to which communication requests are being made. In some systems it may be desirable that each channel should get equal access to physical media bandwidth and that bandwidth should be equally shared amongst all the channels.

On solution to allocating bandwidth amongst the channels is a round robin, one
15 message per channel allocation. However, a message based allocation may unfairly favor requestors with larger messages. For example, those channels with requests for large amounts of data would get more bandwidth than those channels with requests for smaller amounts of data.

Therefore, it would be desirable to have a more fair bandwidth allocation than a
20 pure one message per channel mechanism. It may also be advantageous to group classes of channels together in order to vie for gross amounts or percentages of media bandwidth. This combination of channel grouping into classes and bandwidth allocation based on
25 class may provide for differing levels of service.

Another problem that must be addressed when servicing multiple communication
channels is how to keep track of which channels need serviced and which channel should
be serviced next. One solution is to provide a block of memory in which a single bit
30 maps to a channel. If the bit for a particular channel is set then it indicates that the

corresponding channel has a pending request. After a channel is serviced, this block of memory may be examined to determine the next channel that has a pending request. This solution requires an amount of memory equal to the maximum number of channels supported. For example, if 64K channels are supported then 64K bits of memory are needed to indicate the status of each channel. One drawback of this solution is that much time must be spent scanning the memory for the next channel that has a pending request. For example, if the memory can be searched 32 bits at a time then the worst case search time would be 2K memory accesses (64K bits divided by 32 bits).

Alternatively, linked list structures may be used to indicate the next channel requiring service. The linked list solution avoids the search time inefficiencies of the previously described solution since the next channel to be serviced may be immediately indicated by the linked list structure. However, a linked list solution requires much more memory to implement. The first described solution in which one bit per channel is used to indicate whether or not a request is pending in a channel requires in the order of N bits of memory for N channels. However, a linked list solution requires memory in the order of $N \cdot \log N$. For example, for a 64K channel system a linked list solution would require on the order of 128 kilobytes of memory to maintain a list of active queues if the list was not sparse (approximately 16 times as much memory as the one bit per channel solution).

Thus, while a linked list structure may be time efficient for determining the next queue to be serviced, it is not memory efficient. In contrast, the one bit per channel solution may be somewhat more memory efficient, but may not be time efficient for determining the next queue to be serviced. Thus, it would be desirable to have a solution for determining which queue is to be serviced that is efficient in both time and memory.

Thus, servicing multiple communication channels evokes numerous problems. It is desirable to select the next channel or queue for servicing in a timely fashion. It may also be desirable to select the next channel or queue for servicing in a fair manner while providing for different service classes. It may also be desirable to perform the

	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---

SUMMARY OF THE INVENTION

Many of the problems discussed above may be alleviated by employing a hierarchical channel map or bitmap to determine when communication channels should be serviced. Each bit in a higher level in a hierarchical channel map indicates that at least one bit is set in a corresponding group of bits in the next lower level of the hierarchical channel map. Each higher level bit may provide this indication for a fixed sized group of lower level bits. At the lowest level of the hierarchical channel map, each bit maps to one of the communication channels. If a communication channel has a pending request, then the corresponding bit is set in the lowest level. If a communication channel no longer has any pending requests, its corresponding bit is cleared at the lowest level. Changes to the lowest level of the hierarchical channel map may be propagated up the hierarchy so that each bit at a higher level reflects whether or not any of the bits in the corresponding lower level group are set.

At each level of the hierarchical channel map a service mask is provided to track which bits have already been traversed at that level. A service mask may be cleared once all bits within a current bit group have been traversed. In an alternate embodiment, a different service mask may be provided per level and per service class to allow channels to be divided into different service classes. Sections of channels may be assigned to different service classes according their grouping at the highest level of the hierarchical channel map. A service array may indicate an order in which each service class receives service bandwidth where the order indicated by the service array is continually repeated. For the current service class indicated by the service array, the hierarchical channel map is examined according to the positions indicated by the service masks for the current service class.

In one embodiment a system for servicing a plurality of communication channels may include a host adapter configured to service communication requests wherein the

host adapter supports up to a maximum number of communication channels for servicing the requests. A communication fabric may be configured to provide a limited bandwidth to the host adapter for servicing the communication requests. A memory may be configured to store a lowest level of a hierarchical channel map. The lowest level may include one bit for each communication channel wherein each bit of the lowest level is set to indicate that the communication channel to which it is mapped has a pending communication request and is cleared if there is no pending communication request. Memory may also be configured to store a top level of the hierarchical channel map. Each bit of the top level maps to a section of the lowest level and is set if at least one bit in the section of the lowest level to which it is mapped is set and is cleared if not of the bits in the lowest level section to which it is mapped are set. The host adapter may determine a next channel to be serviced by examining or searching the hierarchical channel map.

In an embodiment, a method for servicing a plurality of communication channels may include selecting a set bit in a top level of a hierarchical channel map wherein the set bit indicates a group of bits in a next level of the hierarchical map to be examined. The method may include examining in each of one or more intermediate levels of the hierarchical map only a group of bits as indicated by the set bit selected in the previous level, and selecting a set bit from each examined group. A lowest level may be examined wherein only the group of bits indicated by the previous intermediate level need be examined. The method may include selecting a bit from the examined lowest level group wherein the selected lowest level bit indicates one of the communication channels to be serviced.

Generally speaking a system for servicing communication queues may include memory configured to store a hierarchical channel map having a plurality of levels wherein each bit of the lowest level is mapped to a different one of a plurality of communication channels and wherein each bit of each higher level is mapped to a group of bits at the next lower level. The system may include a host adapter configured to

maintain the hierarchical bitmap wherein each bit at the lowest level is set if the channel to which it is mapped has a pending communication request and is cleared if not. Each bit of each higher level is set if at least one bit is set in the lower level group to which is mapped and cleared if not. The host adapter may be configured to examine the

5 hierarchical bitmap in order to determine a next one of the communication channels to service.

Attorney's Office
of the
United States
Patent and
Trademark
Office
Washington, D.C.
20540

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

Figure 1 is an illustration of a computing system in which numerous communication channels are supported;

Figure 2 is a conceptual illustration of queuing communication requests for different communication channels;

Figure 3 is an illustration of a hierarchical bitmap for indicating which ones of a plurality of communication channels has a pending communication request needing service;

Figure 4 is an illustration of a hierarchical bitmap such as illustrated in Figure 3 along with a service mask register for each level of the hierarchical bitmap;

Figure 5 illustrates a flow diagram of a method for locating a next channel to be serviced;

Figure 6 is an illustration of a memory map showing how a hierarchical bitmap may be mapped into a block of memory;

Figure 7 illustrates a pseudo-code flowchart showing a method for locating a next channel queue to be serviced;

Figure 8 is an illustration of a hierarchical bitmap such as illustrated in Figure 3 along with a service mask register for each of four service classes and for each level of the hierarchical bitmap; and

5 Figure 9 shows an illustration of service class bitmaps for a four service class example.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will
10 herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

15

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Turning now to Figure 3, a hierarchical bitmap 200 is illustrated for indicating which ones of a plurality of communication channels has a pending communication request needing service. The embodiment illustrated in Figure 3 supports up to 64K communication channels. However, any number of communication channels may be supported by the hierarchical bitmap mechanism described herein. In the 64K channel example of Figure 3, the lowest level (level three) of the hierarchical bitmap 200 includes 64K bits where each bit represents whether or not a corresponding channel requires service. For example, if a particular bit is set in level three of the hierarchical bitmap 200 then the corresponding channel has a pending communication request. If a particular bit is cleared in level three of hierarchical bitmap 200, then the corresponding communication channel does not have a pending communication request. The status indicating bits of level three are divided into multiple channel groups 202. In one example of hierarchical bit map 200, each channel group 202 includes 32 bits representing 32 different channels. Thus, for the 64K channel example, level three would include 2K channel groups 202 (64K bits, each one mapping to a different channel). A 32-bit size for channel groups 202 is merely one example. Different sizes may be employed for channel groups 202. It may be preferable that channel groups 202 are sized according to the maximum number of bits that can be read and written by a single memory access.

The next higher level (level two) of hierarchical bitmap 200 includes one bit for each channel group 202 of the next lower level (level three). The bits of level two are also divided into 32-bit groups 204. Thus, for the example in Figure 3, level two would include 2K bits (64K/32), one for each of the 2K channel groups 202 of level three. Bit 1 of level two may map to bits 1-32 of level three, bit 2 of level two may map to bits 33-64 of level three, etc. Thus, level two includes 64 level two groups 204. Each bit in level two of the hierarchical bitmap 200 indicates whether or not a bit in a corresponding channel group 202 of level three is set. For example, if a particular bit is set in level two

then that indicates that at least one bit of the 32 bits in the corresponding level three group 202 is set. Preferably the level two groups 204 are divided into a group size that may be read or written by a single memory or register access.

5 The top level (level one) of hierarchical bitmap 200 includes one bit corresponding to each of the level 2 groups 204. In the example of Figure 3, level one would include 64 bits, each one corresponding to a different one of the 64 level 2 groups 204. If a particular bit is set in level one, then that indicates that one of the bits in the corresponding level 2 group 204 is also set, which in turn indicates that at least one bit is
10 set in the level three channel group 202 corresponding to the bit set in the level 2 group 204. Note that while the hierarchical bitmap 200 in Figure 3 has been illustrated with each higher level bit of the bitmap hierarchy representing 32 bits at a lower level, different group sizes may be employed and the group sizes may differ between levels. Also, while three levels have been illustrated for the example of Figure 3, different
15 numbers of levels (e.g., two, or more than three) may also be employed.

 Note that the hierarchical bitmap 200 requires only slightly more memory than a flat memory map of the communication channels. For example, for 64K channels a flat (single level) map would require 64K bits of memory. A hierarchical bitmap, such as in
20 Figure 3, would require 64K bits at level three plus 2K bits at level two plus 64 bits at level one. Thus, the hierarchical bitmap would require approximately 8.25 kilobytes of memory compared to 8 kilobytes for the flat map. Therefore the hierarchical bitmap of Figure 3 requires only slightly more memory than a flat map solution and still far less memory than a linked list solution.

25

 The hierarchical bitmap also provides a fast mechanism for locating the next channel requiring service. Assuming that 32 bits are accessed at a time, the worst case number of memory accesses in one embodiment to the hierarchical bitmap for the 64K bitmap example of Figure 3 is four – two at the first level to locate a set bit out of the 64
30 bits of the first level followed by one access to the level two group 204 corresponding to

the set bit of level one, followed by one access to the channel group 202 of level three corresponding to a set bit of the accessed level two group 204. In other embodiment in which a current lowest level is first read to find a next set bit, two additional reads may be required.

5

In a preferred embodiment, the level one bits may be stored in one or more registers as oppose to random access memory so that level one accesses may be performed much faster. However, the exact memory implementation is not critical and, if resources permit, all levels may be stored in a register file, or on chip memory, or SRAM, or another fast memory, or any other type of memory.

10

After a channel has been serviced, the bit corresponding to that channel in level three is cleared if the channel does not have any more pending requests. If the clearing of a bit in a channel group 202 results in all bits of that group cleared, then the corresponding bit in level two must also be cleared, which may also require a bit in level one being cleared if all the bits in the level two group are cleared. Similarly, as bits are set in level three to indicate pending channel requests, corresponding bits in level two and level one may need to be set. The clearing and setting of bits throughout the hierarchical bit map may be referred to as maintaining the validity of the hierarchical bitmap. However, at the worst case only three reads and three writes are needed to set a low level bit and its corresponding hierarchical bits or to clear a bit. For example, when a communication request has been serviced and no further requests are pending for that channel, then the corresponding channel group 202 is read and modified to reflect the now clear channel. If no other bits are set in the channel group 202, then the corresponding level two group 204 is read and modified to indicate that the level three channel group is all clear. If no other bits are set in the corresponding level two group 204, then the corresponding portion of level one is read and modified to indicate that the level two group is all clear. Note that this sequence of three reads and three writes to set or clear a bit is the worst case. In many situations only one or two pairs of reads and writes will be needed to set or clear a bit. This is because when a bit is set or cleared

15

20

25

30

other bits in that group may also be set so that no change is needed at the next higher level.

In one embodiment, the hierarchical bitmap is traversed in a round robin manner to ensure fairness in bandwidth allocation amongst the channels. To further ensure fairness, channel service may not be based on message size, but instead upon a finer fixed granularity. This is to prevent channel requests with a large message size from receiving a higher portion of bandwidth than channel requests with a small message size. A smaller service unit, such as a cell, frame, flit or quantum, may be used for the channel service size. The terms quantum or service unit will be used hereinafter to indicate the access time or size allowed to service each channel. In other words, each channel with a pending request is allowed to access the interconnect fabric or communication media for one quantum before it must relinquish the communication media to another channel. In a round robin protocol, a given channel is not allowed repeat access to the communication media until all other channels with pending requests have had access to the media.

Turning now to Figure 4, a hierarchical bitmap such as illustrated in Figure 3 is shown along with a service mask register for each level of the hierarchical bitmap. The service mask registers are used to keep track of which bits have been traversed at a given level for round robin servicing. For 32-bit hierarchical bitmap groups, each service mask may be a 32-bit bitmask or a 5-bit position pointer. Each service mask indicates the position of the most recently traversed bit for an active group in that level. When the active group in a given level is examined to determine the next bit indicating that service is required, the bits are examined starting at the next highest bit in the group from the position indicated by the service mask for that level. When the end of a group is reached, the service mask is cleared and the next group is traversed from the beginning.

Turning now to Figure 5, one example of a method for locating the next channel to be serviced is illustrated. Upon an initial access all service masks are cleared, as indicated at 300. The first set bit is located in the top level of the hierarchical bitmap, as

indicated at 302. Service mask one is then updated to indicate the position of the first set bit in level one of the hierarchical bitmap, as indicated at 304. The level two group indicated by the bit set in level one is then accessed, as indicated at 306. The accessed level two group is then examined to locate a set bit, as indicated at 308, and the level two service mask is updated to indicate the position of a set bit, as indicated at 310. Then the level three group indicated by the set bit in level two is accessed, as indicated at 312, and the first set bit is located in the level three group, as indicated at 314. The level three service mask is updated to indicate the position of the set bit in level three of the hierarchical bitmap, as indicated at 316.

The corresponding channel queue indicated by the identified set bit in level three is then serviced for one quantum, as indicated at 318. For subsequent accesses the current level three hierarchical bit group is examined to locate the next set bit after the position indicated by the level three service mask, as indicated at 314. If no more bits are set after the position indicated by the level three service mask, then the current level two group is examined to locate the next set bit after the position indicated by the level two service mask, as indicated at 308. If no more bits are set in the level two group, then the level one group is examined to find the next set bit, as indicated at 302.

Turning now to Figure 6, a memory map is illustrated showing how a hierarchical bitmap may be mapped into a block of memory. The memory for storing the hierarchical bitmap may be part of the system memory of a computer system or may be a dedicated memory. The memory map illustrated in Figure 6 assumes that the top level of the hierarchical bitmap is stored in one or registers of a host adapter or other controller. Thus, only level two and three of the hierarchical bitmap are mapped to memory. Figure 6 illustrates an example of a 64K channel hierarchical bitmap, such as shown in Figure 3. Each memory location 1 through 64 stores one of the level two groups of the hierarchical bitmap. Each memory location is 32 bits wide. The next 2K memory locations store the level three channel bit groups. Each address holds a 32-bit level three bit group that corresponds to a level two bit.

Turning now to Figure 7, a pseudo-code flowchart is illustrated showing a method for locating a next channel queue to be serviced. The method is similar to that illustrated in Figure 5, except it is shown in pseudo-code and corresponds to the memory map of Figure 6. SM[1:3] correspond to the service mask registers for levels 1-3. SM[1] holds values from 1 to 64 corresponding to a bit set in the level one bit mask group. SM[2:3] holds values from 1-32 for the current working level two and level three bit positions. BM [1:3] are the current working level one to three hierarchical bit mask groups. The function firstbit(BM, SM) is a function that returns the first bit set in the BM, after the bit position indexed by the SM. A zero indicates that no bits are set. The function memory(A) returns the value and memory at location A. The memory holds the hierarchical bitmaps, e.g., as shown in Figure 6.

The hierarchical bitmap mechanism has been described in regard to Figures 4-7 as employing a round robin servicing mechanism in which each channel receives one quantum of access service (if needed) for each round through the hierarchical bitmap. An additional mechanism may be employed to provide for different service classes in which some channels may receive a larger portion of the communication media bandwidth than other channels. This may be desirable for different service priorities or differing delay variabilities. It may be that certain channels must be guaranteed a higher portion of bandwidth, or certain channels while not necessarily requiring a large amount of bandwidth may require a very low latency between access, such as in real-time applications. Thus, it may be desirable to provide for different service classes among the channels.

A mechanism may be provided to group the channels into different service classes according to the top level channel groupings of the hierarchical bitmap. As can be seen in Figure 3, each bit at the top level (level one) bitmap represents a fixed set of 1,024 channels, for the 64K channel example. This grouping may be taken advantage of to group channels into different service classes, 1,024 channels at a time. In other words,

channels may be divided into different service classes according to their grouping at the top level of the hierarchical bitmap. In the 64K channel example, this would provide for up to 64 different service classes. In practice, fewer service classes may be needed. Therefore, a lesser number of service classes may be provided for, such as eight.

5

In one embodiment, a service class mask may be used for each service class to indicate which of the 1,024 channel sections are members of that service class. For the example of eight service classes for the 64K channel example, this would require eight 64-bit service class masks. Each one of the eight 64-bit service class masks would
10 indicate which of the 1,024 channel chunks are members of that service class. Note that other structures may be used besides mask registers to indicate which channel sections are members of a particular service class.

In addition, service mask registers, such as shown in Figure 4, may be duplicated
15 for each service class, as indicated in Figure 8. Figure 8 illustrates four different service classes. A service array (also shown in Figure 8) is used to determine how the service classes are serviced. A service array includes X number of entries, where X determines the bandwidth granularity that may be assigned to different service classes. Each entry in the service array indicates the particular service class that should be serviced during the
20 current quantum. At the end of the current quantum, the next entry in the service array is visited. When the end of the service array is reached, the order is repeated from the beginning of the service array. Thus, the service array may be viewed as a circular indication of the order in which service classes receive a quantum of bandwidth. For example, if a service array contains 64 entries the bandwidth may be partitioned into 1/64
25 divisions (approximately 1.5% of the total bandwidth). Each time a service class is indicated in the service array, it receives an additional 1.5 percent of the bandwidth. The service array and the mapping of channel sections to service classes may be configured by software to achieve the desired bandwidth apportionments for different channels and different service classes.

30

Refer now to Figure 8 for an example of four service classes and an eight entry service array in the 64K channel hierarchical bitmap example. This configuration gives 12.5 percent bandwidth granularity. The example of Figure 8 would give class four 50 percent of the media bandwidth, class one 25 percent, and classes three and four each with 12.5 percent. In the first quantum, a channel from service class four gets access. In the next quantum, a channel from service class one gets access. Then from service class four, two, four, one, four, three, and the cycle then repeats. For each service class, a method such as illustrated in Figure 5 is used to locate the next channel requiring service. However, the service mask set corresponding to the particular service class is used. Thus, only bit groups within the hierarchical bitmap corresponding to channels within the current service class as indicated by the service array are examined to locate the next channel to be serviced.

Figure 9 shows an illustration of service class bitmaps for a four service class example. As shown in the example of Figure 9, the first two channel sections (channel number 1-2,048) are assigned to service class one. The next six channel sections are assigned to service class two. The next four are assigned to class three, and the last channel section is assigned to service class four. As can be seen from the example of Figure 9, not all channels need be assigned to a service class, e.g., if those channels will not be used. Also, note that the service class bitmaps need not be contiguous. For example, channel sections 64 and 63 could be assigned to the same service class as channel sections 3-8. Also, note that the same bits may be set in different service class bitmaps so that the same channel sections may be assigned to more than one service class. For instance, it may be desirable to make channels 1-1,024 a member of every class, giving those channels equal access to all the channel bandwidth. This may be useful for channels that don't require much bandwidth but need to limit service latency. In such applications, certain channels may limit their bandwidth needs at the source and thus may not require a large amount of bandwidth. However, when access is needed, by being a member of every service class, such a channel may be provided quicker access to the communication media (e.g. reducing send jitter) when they need it.

In a preferred embodiment if a given service class does not have any channels with a pending request then the next entry in the service array is visited immediately. In the example described above, channels are divided into 1,024 channel sections at the top level of the hierarchical bitmap and thus assigned to service classes 1,024 channels at a time. If a finer granularity than 1,024 channels is desired for service class grouping, then the structure of the hierarchical bitmap may be changed. For example, for a 64K channel architecture, the level one bitmap may be 128 bits mapping to 16-bit groups at level two which map to 32-bit groups at level three. This would provide a granularity of 512 channel queues for being divided into service classes.

The hierarchical bitmap method for servicing multiple communication channels may be applied to virtually any application in which multiple communication channels share a limited bandwidth. For example, the hierarchical bitmap mechanism may be implemented in a system such as illustrated in Figures 1 and/or 2. The hierarchical bitmap mechanism may be implemented largely in software. In one embodiment a host channel adapter is configured to implement the hierarchical bitmap. As discussed above, higher levels of the bitmap may be implemented in registers within a host adapter integrated circuit and the larger levels of the bitmap may be implemented in a dedicated memory or a system memory. Service masks and service class masks may be implemented as registers within such an integrated circuit. Various channel requests queues may also be supported in the host adapter integrated circuit. Channel service logic may be included to determine the next channel to be serviced as described above.

Various modifications and changes may be made as would be obvious to a person skilled in the art having the benefit of this disclosure. It is intended that the following claims be interpreted to embrace all such modifications and changes and, accordingly, the specifications and drawings are to be regarded in an illustrative rather than a restrictive sense.

WHAT IS CLAIMED IS:

- 1 1. A system for servicing a plurality of communication channels, comprising:
2
3 a host adapter configured to service communication requests from communication
4 requestors to communication targets, wherein said host adapter provides
5 for up to a maximum number of communication channels to service the
6 communication requests;
7
8 a communication fabric configured to provide a limited bandwidth to said host
9 adapter to service said communication requests, wherein said
10 communication fabric couples said communication targets to said host
11 adapter;
12
13 a first memory configured to store a lowest level of a hierarchical channel map,
14 wherein said lowest level comprises one bit for each communication
15 channel supported by said host adapter, wherein each bit of said lowest
16 level is set to indicate that the communication channel to which it is
17 mapped has a pending communication request and is cleared if the
18 communication channel to which it is mapped does not have a pending
19 communication request;
20
21 a second memory configured to store a top level of the hierarchical channel map,
22 wherein each bit of said top level maps to a section of said lowest level,
23 wherein each bit of said top level is set if at least one bit in the section of
24 said lowest level to which it is mapped is set and is cleared if none of the
25 bits in the section of said lowest level to which it is mapped are set;
26
27 wherein said host adapter is configured to determine a next channel to be serviced
28 by examining said hierarchical channel map.

1 2. The system as recited in claim 1, wherein said hierarchical channel map further
2 comprises one or more intermediate levels, wherein each intermediate level comprises a
3 plurality of bits, each one mapping to a group of bits in the next lower intermediate level,
4 wherein each bit indicates if at least one bit in the group to which it maps is set, wherein a
5 highest intermediate level comprises groups of bits mapped to individual one of the bits
6 of said top level, and wherein a lowest intermediate level comprises bits each mapping to
7 one of the groups of said lowest level of the hierarchical channel map.

1 3. The system as recited in claim 2, wherein said host adapter is configured to
2 examine one group of bits at one or more levels of the hierarchical bit map to identify a
3 next channel to be serviced.

1 4. The system as recited in claim 3, further comprising a service mask for each level
2 of the hierarchical channel map, wherein each intermediate level service mask indicates
3 the next bit position to be examined within a selected group of bits to determine a
4 selected group of bits to be examined at the next level, wherein the service mask for the
5 lowest level indicates the next bit position to be examined within a selected group to
6 determine the next channel with a pending request to be services.

1 5. The system as recited in claim 4, wherein said host adapter is configured to
2 service each pending communication request for one service unit before moving on to a
3 next pending communication request, wherein the next pending communication request is
4 determined by examining a current selected group in said hierarchical channel map for
5 the next set bit in order from a position indicated by the corresponding service mask.

1 6. The system as recited in claim 4, wherein said one service unit is less than a
2 maximum message size for the communication requests.

1 7. The system as recited in claim 1, wherein said first memory and said second
2 memory are part of a memory block accessible by said host adapter.

1 8. The system as recited in claim 1, wherein said second memory is a register
2 comprised within an integrated circuit with said host adapter.

1 9. The system as recited in claim 1, further comprising a plurality of service class
2 masks, wherein each service class mask is configured to map a number of said
3 communication channels to one of a plurality of service classes, wherein each service
4 class is allocated a portion of said limited bandwidth on said communication fabric.

1 10. The system as recited in claim 9, further comprising a service array comprising a
2 plurality of entries, wherein each entry indicates one of said service classes to be serviced
3 during a current service unit, wherein said service classes are selected in order according
4 to said entries in said service array, wherein the order is repeated, and wherein the next
5 channel to be serviced is selected from the current service class.

1 11. The system as recited in claim 10, further comprising for each service class a
2 service mask for each level of the hierarchical channel map, wherein when its
3 corresponding service class is being serviced each service mask indicates the next bit
4 position to be examined within a selected group of bits to determine a selected group of
5 bits to be examined at the next level, except wherein the service mask for the lowest level
6 indicates the next bit position to be examined within a selected group to determine the
7 next channel with a pending request to be services.

1 12. A method for servicing a plurality of communication channels, comprising:
2
3 for a first service time:

4
5 selecting a set bit in a top level of a hierarchical channel map, wherein the

6 set bit in the top level indicates a group of bits in a next level of the
7 hierarchical channel map to examine;

8
9 examining in each of one or more intermediate levels of the hierarchical
10 bit map only a group of bits indicated by the set bit selected in the
11 previous level and selecting a set bit from each examined group,
12 wherein said top level is the previous level for a first intermediate
13 level;

14
15 examining in a lowest level of the hierarchical bit map only a group of bits
16 indicated by the previous intermediate level and selecting a set bit
17 from the examined group of the lowest level, wherein the selected
18 bit at the lowest level indicates one of the plurality of
19 communication channels to be serviced; and

20
21 servicing a communication request from the communication channel
22 indicated by the selected bit from the lowest level of the
23 hierarchical channel map.

1 13. The method as recited in claim 12, further comprising:

2
3 for subsequent service times:

4
5 selecting the next set bit from the group of bits in the lowest level from
6 which the previous set bit was selected, or if no more bits are set in
7 that group, selecting the next set bit from the next higher level in
8 the same group as the previous selected bit at that level or
9 repeating said selecting the next set bit from the next higher level
10 until a set bit is found and then selecting a set bit from each group
11 at each lower level as indicated by the set bit at the previous level;

12 and
13
14 servicing subsequent communication requests during each subsequent
15 communication time from the communication channel indicated by
16 the selected bit from the lowest level of the hierarchical channel
17 map for each subsequent service time.

1 14. The method as recited in claim 12, further comprising:

2
3 setting a bit in the lowest level of the hierarchical channel map for each
4 communication channel that has a pending communication request; and

5
6 setting a bit at each higher level of the hierarchical bit map to indicate that a
7 corresponding group of bits in the next lower level has at least one set bit.

1 15. The method as recited in claim 12, further comprising updating a service mask for
2 each level of the hierarchical channel map to indicate the position of the last selected set
3 bit within the indicated group of bits, wherein each time an indicated group of bits is
4 examined to select the next set bit, it is examined starting at the next bit position from the
5 position indicated by the corresponding service mask.

1 16. The method as recited in claim 13, wherein each bit position of the top level of the
2 hierarchical channel map represents a different section of the plurality of communication
3 channels, the method further comprising:

4
5 mapping each section of communication channels to a different service class;

6
7 indicating in a service array an order in which each service class is to be serviced;

8
9 choosing a service class according to said service array; and

10
11
12
13

examining only portions of the hierarchical channel map corresponding to sections of communication channels mapped to the chosen service class to select the next set bit at the lowest level.

1 17. The method as recited in claim 16, further comprising updating a service mask for
2 each level of the hierarchical channel map and for each service class to indicate the
3 position of the last selected set bit within the indicated group of bits and for the
4 corresponding service class, wherein each time an indicated group of bits is examined to
5 select the next set bit, it is examined starting at the next bit position from the position
6 indicated by the corresponding service mask for the current service class.

1 18. A system for servicing communication queues, comprising:
2
3 memory configured to store a hierarchical channel map comprising a plurality of
4 levels, wherein each bit of the lowest level is mapped to a different one of
5 a plurality of communication channels, and wherein each bit of each
6 higher level is mapped to a group of bits at the next lower level;
7
8 a host adapter configured to maintain the hierarchical bit map, wherein each bit at
9 the lowest level is set if the channel to which it is mapped has a pending
10 communication request and is cleared if not, and wherein each bit of each
11 higher level is set if at least one bit is set in the lower level group to which
12 it is mapped and is cleared if not; and
13
14 wherein said host adapter is configured to examine the hierarchical bit map to
15 determine a next one of the communication channels to service.

1 19. The system as recited in claim 18, wherein said host adapter is configured to
2 service for one service unit a channel request from a channel mapped to a set bit at the

3 lowest level of the hierarchical channel map, wherein the set bit is selected by examining
4 a current group of bits at the lowest level of the hierarchical channel map to select a next
5 set bit in that group indicating a channel with a pending request, and if no more bits are
6 set in the current group, examining a current group at the next higher level to select a next
7 set bit and then examining the next lower level group indicated by the selected higher
8 level set bit.

1 20. The system as recited in claim 19, further comprising a service mask for each
2 level of the hierarchical bit map, wherein each service mask is configured to indicate the
3 next bit position in the current group to be examined for a set bit.

1 21. The system as recited in claim 20, wherein each service mask is configured to
2 indicate the bit position within the current group for the corresponding level of the last
3 selected set bit in that group, wherein the host adapter is configured to examine each
4 group for the next set bit after the bit position indicated by the corresponding service
5 mask.

1 22. The system as recited in claim 18, wherein each group of bits at one level of the
2 hierarchical channel map has the same number of bits.

1 23. The system as recited in claim 18, wherein each group of bits at the lowest level is
2 accessible by a single memory access.

1 24. The system as recited in claim 18, wherein the top level of the hierarchical
2 channel map is stored within a register comprised within the same integrated circuit as
3 said host adapter.

1 25. The system as recited in claim 18, wherein said host adapter is configured to
2 service channel requests according to service classes, wherein each channel is mapped to
3 one or more service classes, and wherein the next channel to be serviced is selected by

4 examining only portions of said hierarchical channel map corresponding to channels
5 mapped to a current service class.

1 26. The system as recited in claim 25, wherein the current service class is selected
2 according to an order of service classes indicated by a service array.

1 27. The system as recited in claim 25, further comprising one service mask per level
2 of the hierarchical bit map per service class, wherein each service mask is configured to
3 indicate the next bit position in the current group to be examined for a set bit when the
4 corresponding service class is the current service class.

1 28. The system as recited in claim 19, wherein said one service unit is a quantum
2 smaller than a maximum message size for the channel requests.

1 29. A computer readable medium comprising program instructions, wherein said
2 program instructions are operable to:

3
4 select a first bit set in a current top level group of a hierarchical channel map after
5 a position indicated by a top level service mask;

6
7 set the top level service mask to indicate the position of the selected first set bit in
8 the current top level group;

9
10 in a second level of the hierarchical channel map, access a second level group
11 indicated by the selected set bit from the top level;

12
13 select a first bit set in the accessed second level group after a position indicated by
14 a second level service mask;

15
16 set the second level service mask to indicate the position of the selected bit in the

17 accessed second level group;
18
19 in a bottom level of the hierarchical channel map, access a bottom level group
20 indicated by the selected set bit from the second level;
21
22 select a first bit set in the accessed bottom level group after a position indicated by
23 a bottom level service mask;
24
25 set the bottom level service mask to indicate the position of the selected bit in the
26 accessed bottom level group; and
27
28 service a request from a channel indicated by the selected set bit from the bottom
29 level.

- 1 30. A method for servicing a plurality of communication channels, comprising:
2
3 for a first service time, examining a top level of a hierarchical channel map to
4 select a section of the communication channels in which at least one
5 channel has a pending communication request, wherein said top level
6 indicates for each of a plurality of sections of the communication channels
7 if at least channel of that section has a pending communication request;
8
9 for said first service time, examining a portion of one or more intermediate levels
10 of the hierarchical channel map to select a lowest level group of the
11 communication channels in which at least one channel has a pending
12 communication request, wherein each intermediate level indicates for each
13 of a plurality of groups of the communication channels if at least one
14 channel of that group has a pending communication request, wherein the
15 groups at each intermediate level are sized smaller than at the previous
16 intermediate level, and wherein examining a portion of each intermediate

17 level determines which portion of the next hierarchical channel map to
18 examine; and
19
20 selecting a next communication channel to be serviced from the lowest level
21 group selected by said examining a portion of one or more intermediate
22 levels.

1 31. A system for servicing a plurality of communication channels, comprising:
2
3 a first memory configured to store one or more levels of a hierarchical channel
4 map, wherein said one or more levels comprises a lowest level for storing
5 an indication of which ones of the plurality of communication channels
6 have pending communication requests;
7
8 a second memory configured to store a top level of said hierarchical channel map,
9 wherein the plurality of communication channels are organized in channel
10 sections, and wherein for each channel section said top level indicates if at
11 least one of the communication channels within that section has a pending
12 communication request; and
13
14 a host adapter configured to determine a next channel to service by examining
15 said lowest level in said first memory, wherein said host adapter
16 determines the next channel to service by examining no more of said
17 lowest level than a portion of said lowest level corresponding to one of
18 said channel sections indicated by said top level as having at least one
19 pending communication request.

1 32. The system as recited in claim 31, wherein said one or more levels stored in said
2 first memory further comprises an intermediate level of said hierarchical channel map,
3 wherein said communication channels are further organized into channel groups, wherein

4 for each channel group said intermediate level indicates if at least one of the
5 communication channels within that group has a pending communication request, and
6 wherein said host adapter is further configured to determine the next channel to service
7 by examining no more of said lowest level than a portion of said lowest level
8 corresponding to one of said channel groups indicated by said intermediate level as
9 having at least one pending communication request, wherein each channel section
10 includes a plurality of channel groups.

1 33. The system as recited in claim 32, wherein said top level stored in said second
2 memory is further configured to indicate which portions of said intermediate level contain
3 at least one indication that at least one channel within one of said channel groups has a
4 pending communication request.

1 34. The system as recited in claim 31, wherein said lowest level comprises a plurality
2 of low level bits, wherein each low level bit is set if a corresponding one of said plurality
3 of channels has a pending communication request and is cleared if the corresponding one
4 of said plurality of channels does not have a pending communication request.

1 35. The system as recited in claim 34, wherein said one or more levels stored in said
2 first memory further comprises an intermediate level comprising a plurality of
3 intermediate level bits, wherein each intermediate level bit corresponds to a group of low
4 level bits, and wherein each intermediate level bit is set if its corresponding group of low
5 level bits has at least one bit set and is cleared if no bits are set in its corresponding group
6 of low level bits.

1 36. The system as recited in claim 35, wherein said top level comprises a plurality of
2 top level bits, wherein each top level bit corresponds to a group of intermediate level bits,
3 and wherein each top level bit is set if its corresponding group of intermediate level bits
4 has at least one bit set and is cleared if no bits are set in its corresponding group of
5 intermediate level bits.

1 37. The system as recited in claim 31, wherein said first memory and said second
2 memory are part of a memory block accessible by said host adapter.

1 38. The system as recited in claim 31, wherein said second memory is a register
2 comprised within an integrated circuit with said host adapter.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

ABSTRACT OF THE DISCLOSURE

A system for servicing communication queues may include memory configured to store a hierarchical channel map having a plurality of levels wherein each bit of the lowest level is mapped to a different one of a plurality of communication channels and wherein each bit of each higher level is mapped to a group of bits at the next lower level. The system may include a host adapter configured to maintain the hierarchical bitmap wherein each bit at the lowest level is set if the channel to which it is mapped has a pending communication request and is cleared if not. Each bit of each higher level is set if at least one bit is set in the lower level group to which is mapped and cleared if not. The host adapter may be configured to examine the hierarchical bitmap in order to determine a next one of the communication channels to service. At each level of the hierarchical channel map a service mask may be provided to track which bits have already been traversed at that level. Multiple such service masks may be provided at each level to support different service classes.

1/8

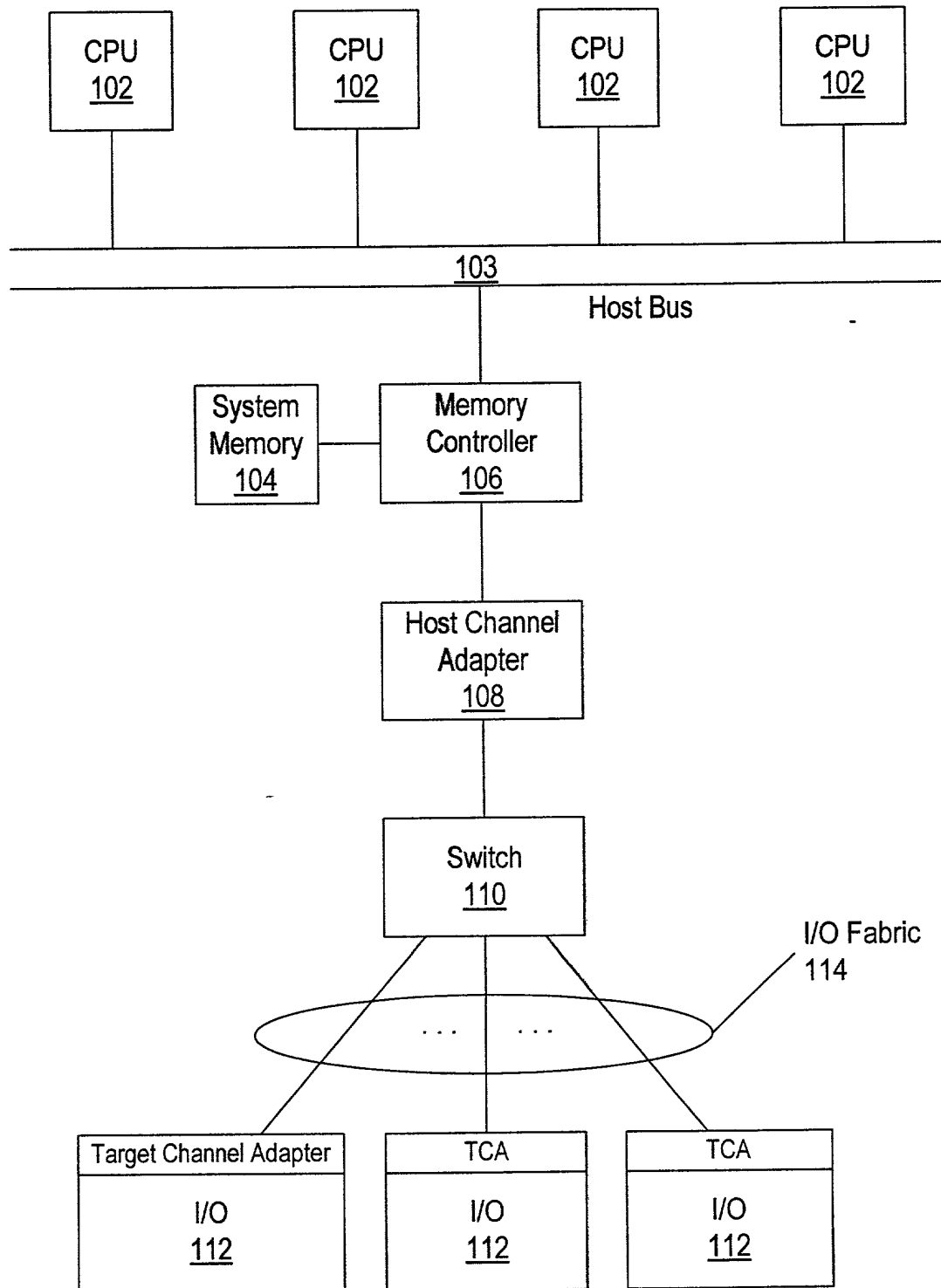


FIG. 1

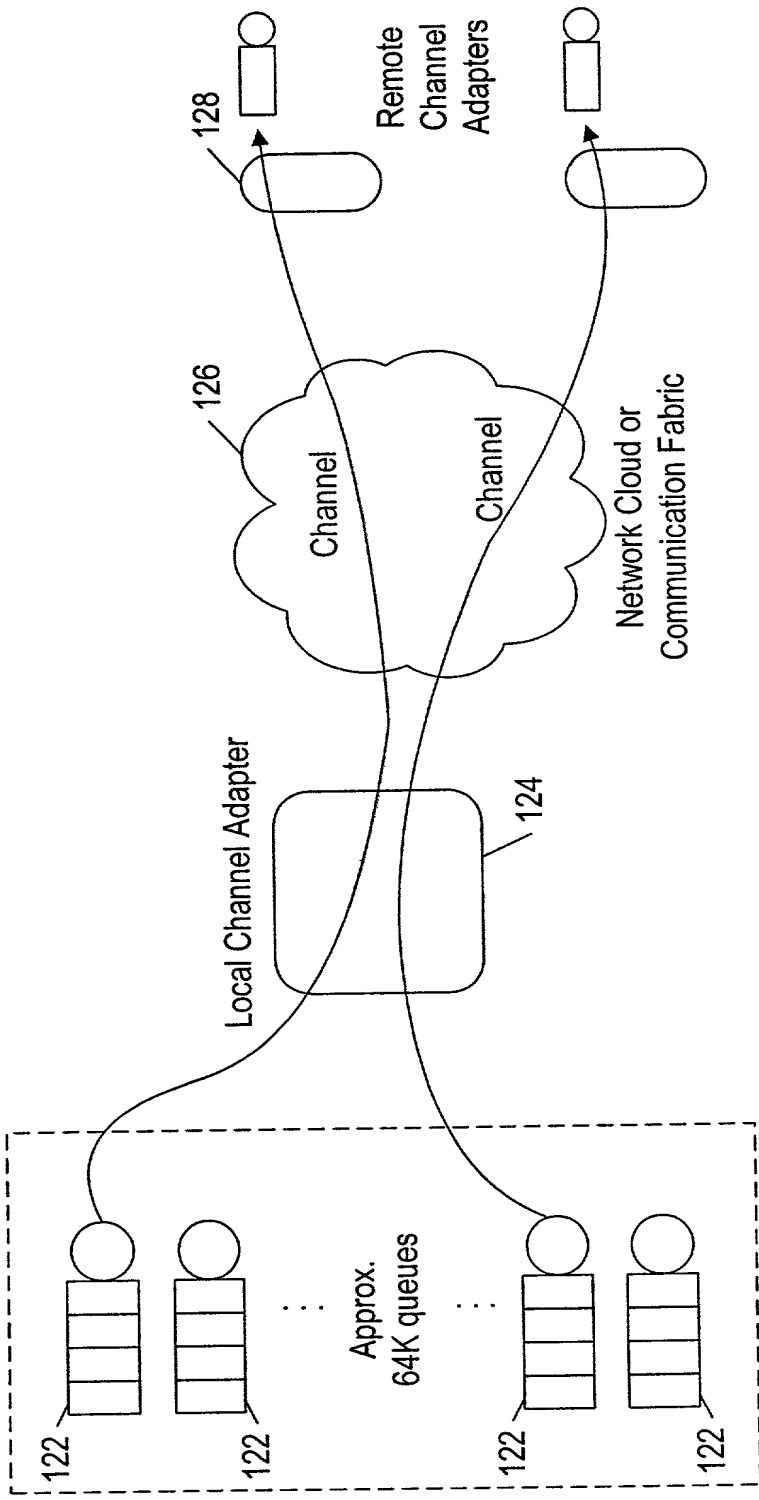


FIG. 2

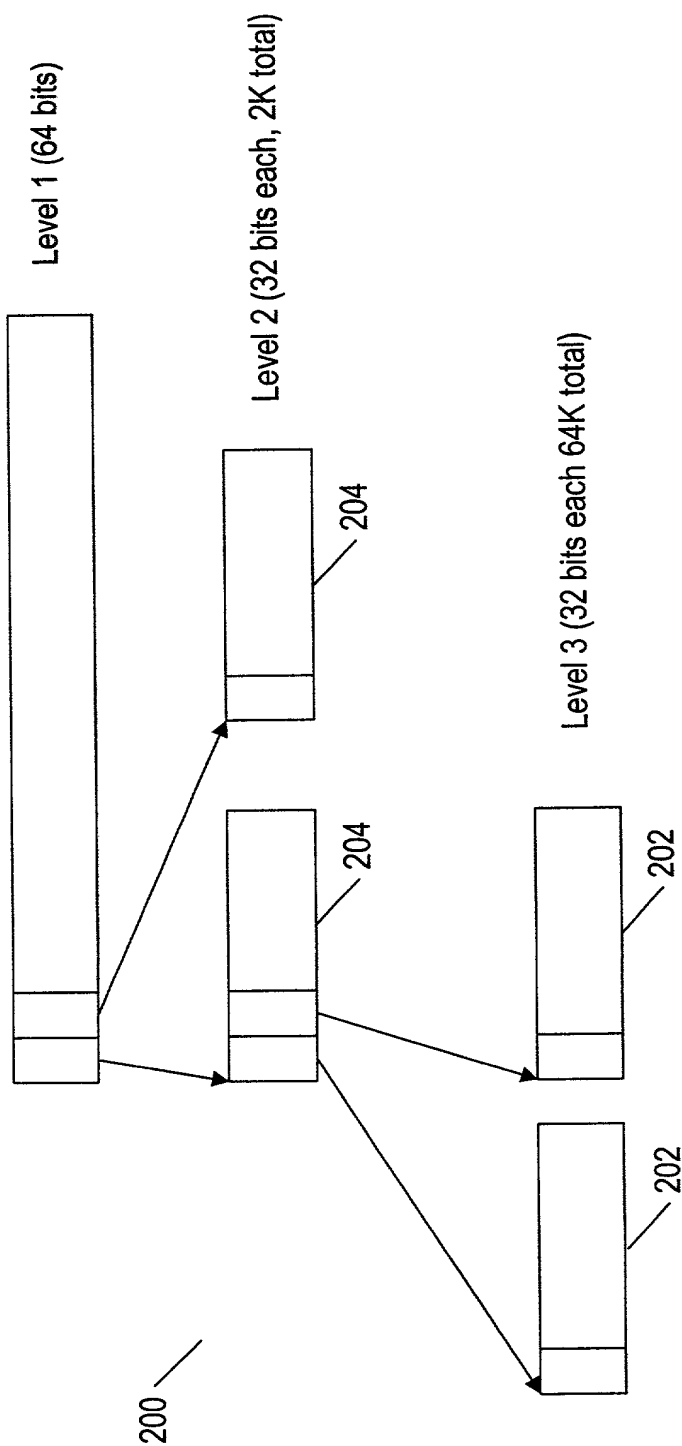


FIG. 3

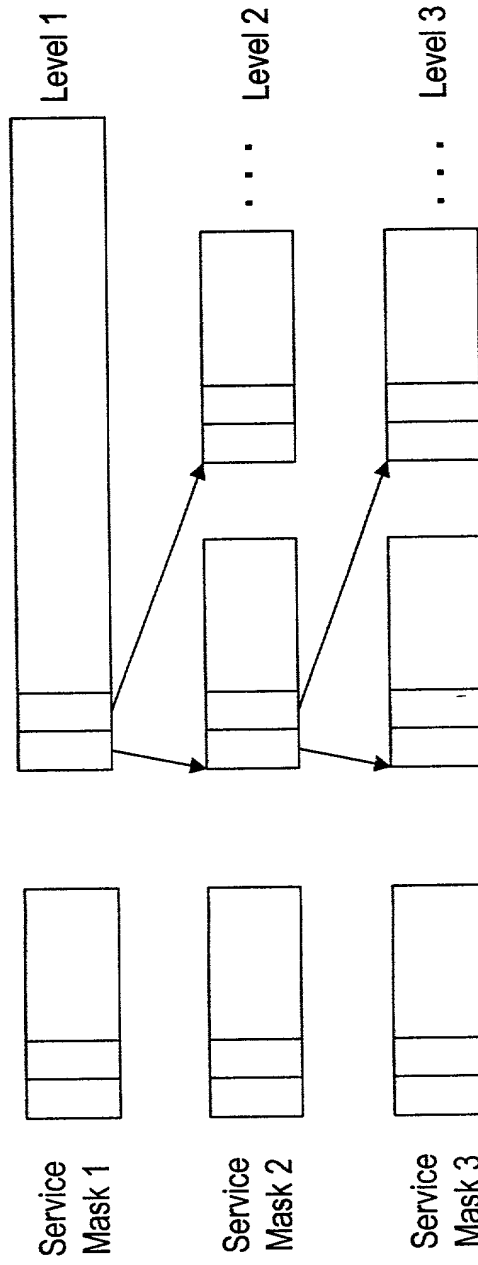


FIG. 4

5/8

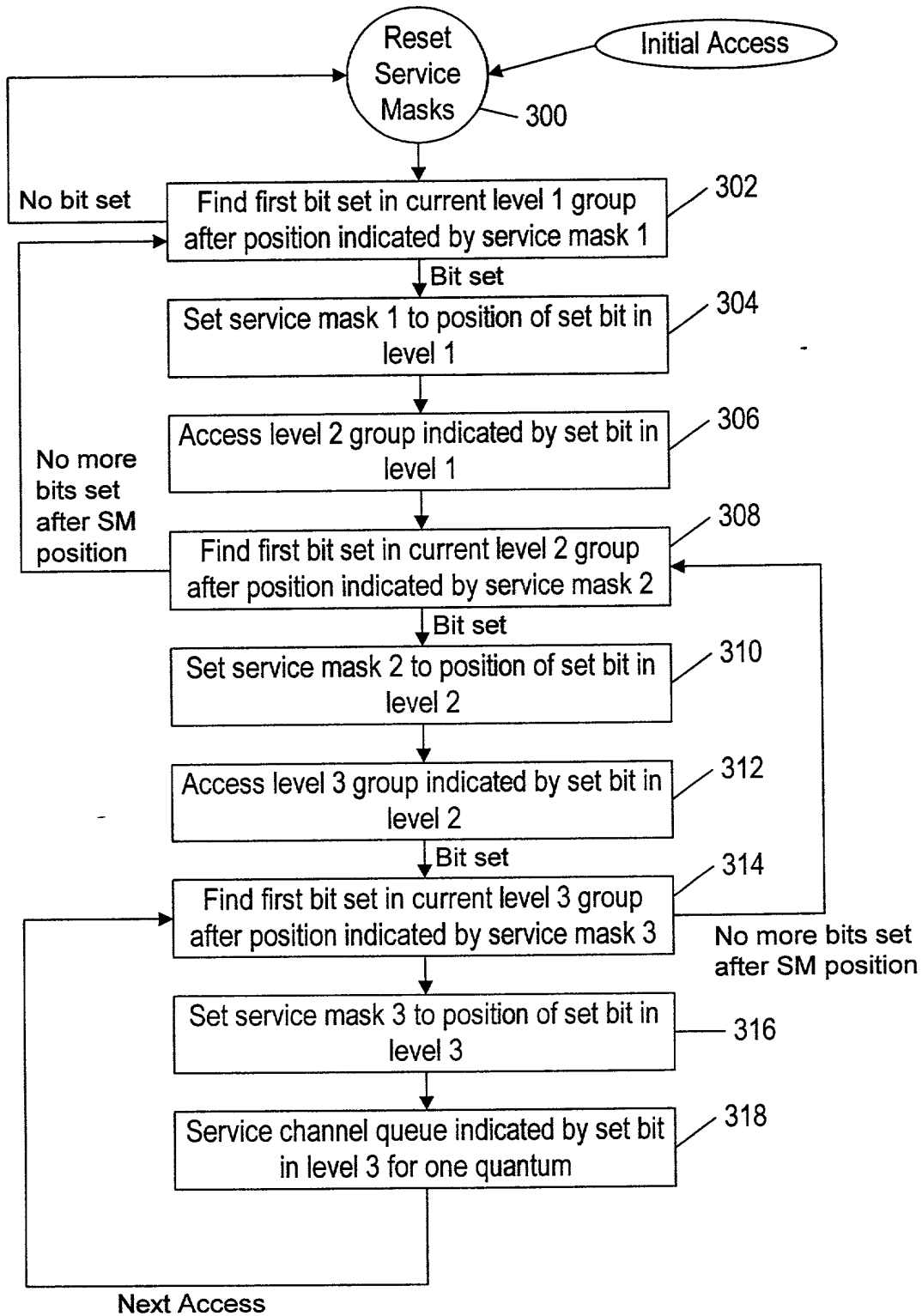


FIG. 5

6/8

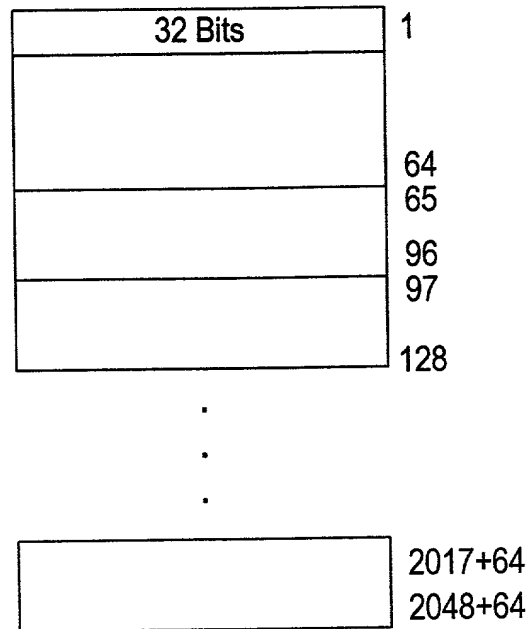


FIG. 6

Class 1 bit map 0x0000000000000003 (channel numbers 1-2048)
 Class 2 bit map 0x00000000000000FC (numbers 2049-8K)
 Class 3 bit map 0x00000000000000F0 (8K-12K)
 Class 4 bit map 0x8000000000000000 (63K-64K)

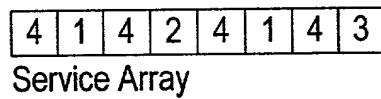


FIG. 9

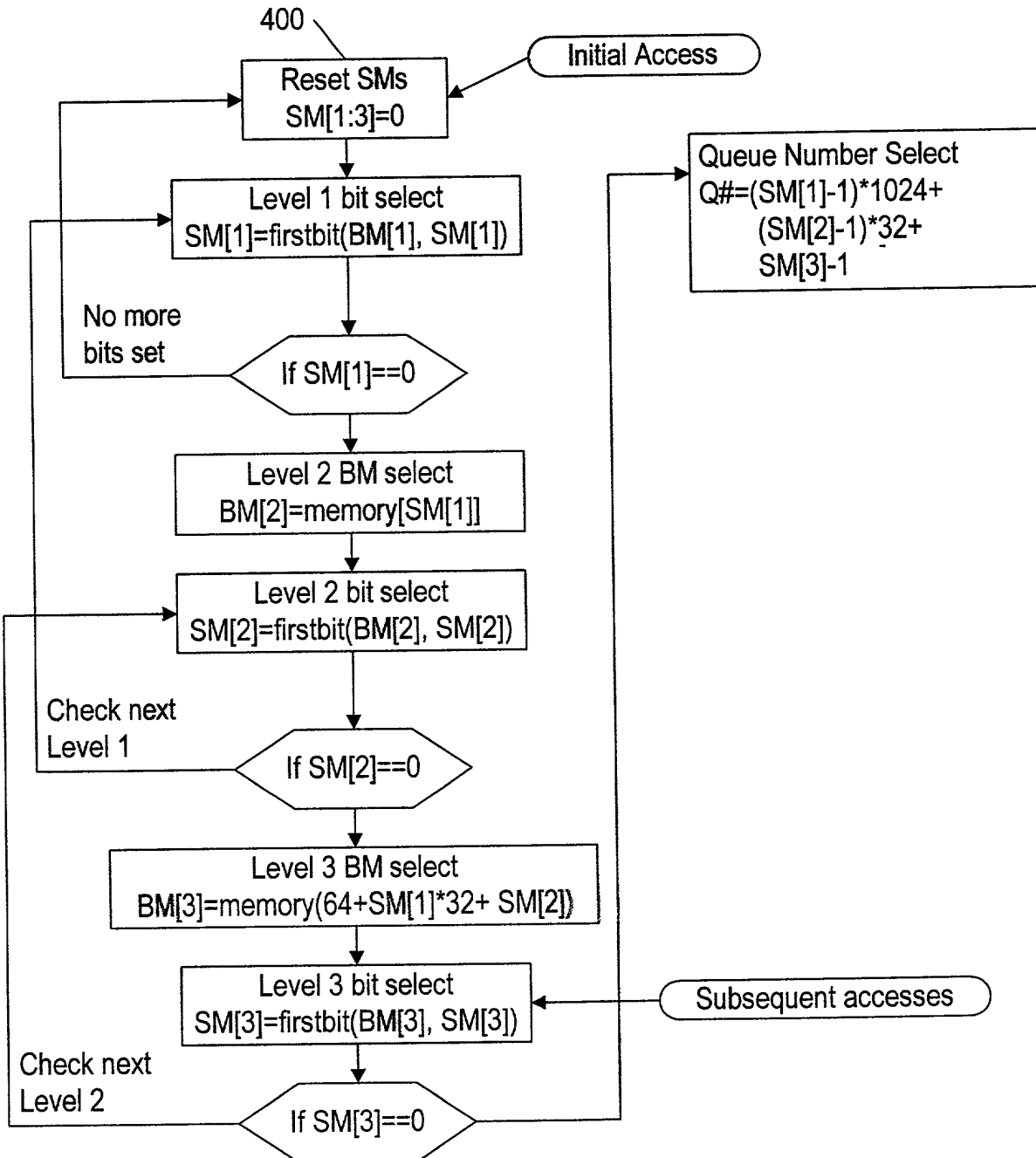


FIG. 7

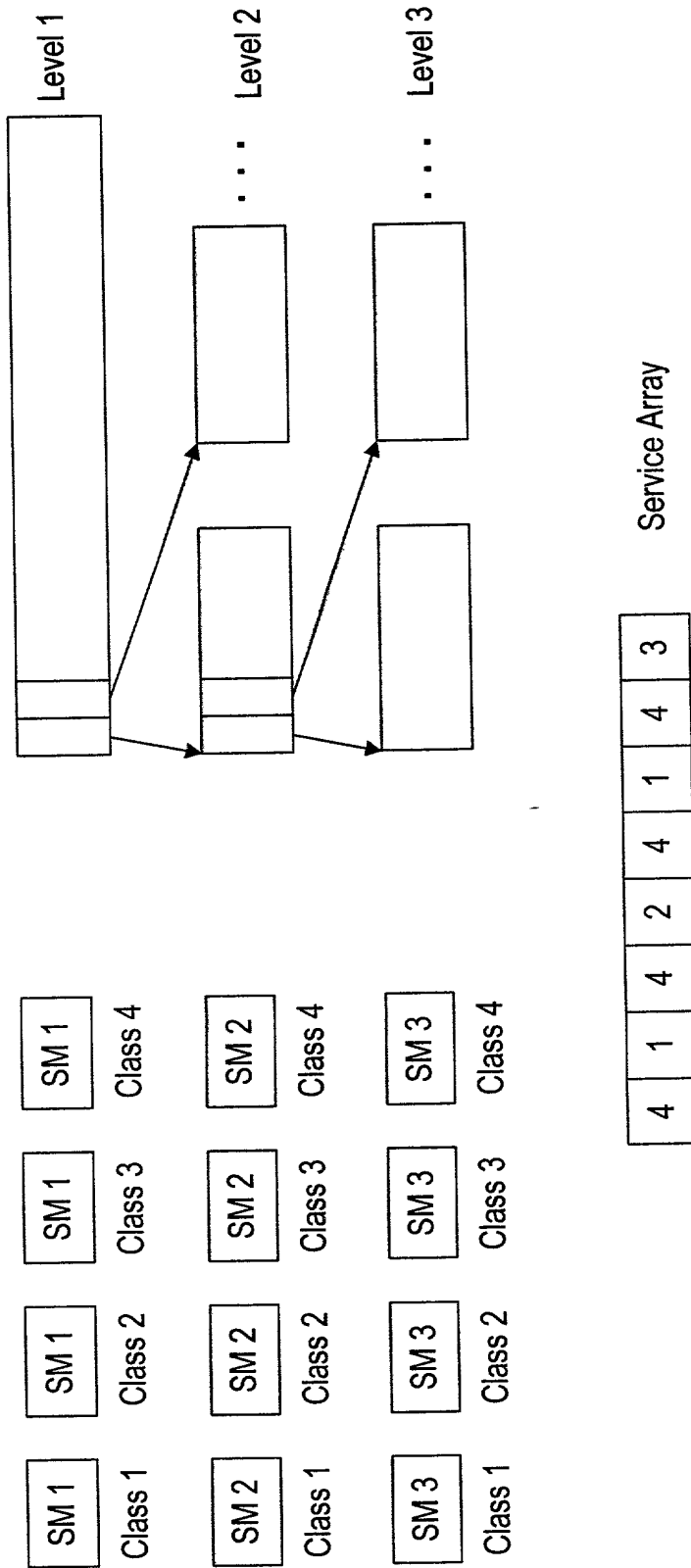


FIG. 8

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled "**MECHANISM FOR EFFICIENT SCHEDULING OF COMMUNICATION FLOWS**," the specification of which:

- ☒ is attached hereto.
☐ was filed on _____ as Application Serial No. _____
and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the Patent and Trademark Office all information known to me to be material to patentability of the subject matter claimed in this application, as "materiality" is defined in 37 C.F.R. § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate listed below, or under § 365(a) of any PCT international application listed below designating least one country other than the United States of America, and have identified below any foreign application for patent or inventor's certificate, or of any PCT international application, having a filing date before that of the application on which priority is claimed.

<u>Prior Foreign Application No.</u>	<u>Country</u>	<u>Filing Date</u> <u>(mm/dd/yy)</u>	<u>Priority</u> <u>Claimed</u>	<u>Cert. copy</u> <u>Attached</u>
N/A				

I hereby claim the benefit under 35 U.S.C. § 119(e) of any United States provisional application(s) listed below.

<u>Provisional Application No.</u>	<u>Filing Date</u> <u>(mm/dd/yy)</u>
N/A	

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s) listed below, or under § 365(c) of any PCT international application listed below designating the United States of America, and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose all information known to me to be material to the patentability of the subject matter claimed in this application, as "materiality" is defined in 37 C.F.R. § 1.56, which became available between the filing date of the prior application and the national or PCT international filing date of this application.

<u>Parent Application No.</u>	<u>Filing Date</u> <u>(mm/dd/yy)</u>	<u>Parent Patent No. (if applicable) or Status</u>
N/A		

I hereby revoke any previous Powers of Attorney and appoint

Kenneth Olsen	Reg. No. 26,493
Timothy J. Crean	Reg. No. 37,116
Joseph T. FitzGerald	Reg. No. 33,881
Robert S. Hauser	Reg. No. 37,847
Alexander E. Silverman	Reg. No. 37,940
Christine S. Lam	Reg. No. 37,489
Anirma Rakshpal Gupta	Reg. No. 38,275
Sean P. Lewis	Reg. No. 42,798
Michael J. Schallop	Reg. No. 44,319
Bernice B. Chen	Reg. No. 42,403
Kenta Suzue	Reg. No. 45,145
Noreen A. Krall	Reg. No. 39,734
Richard J. Lutton, Jr.	Reg. No. 39,756
Monica D. Lee	Reg. No. 40,696
Marc D. Foodman	Reg. No. 34,110

each of said attorneys being employed by Sun Microsystems; and

Dan R. Christen	Reg. No. 39,943
Gentry E. Crook	Reg. No. 44,633
Kevin L. Daffer	Reg. No. 34,146
Mark R. DeLuca	Reg. No. 44,649
Jeffrey C. Hood	Reg. No. 35,198
Robert C. Jahnke	Reg. No. 44,800
B. Noël Kivlin	Reg. No. 33,929
Christopher P. Kosh	Reg. No. 42,760
Robert C. Kowert	Reg. No. 39,255
Lawrence J. Merkel	Reg. No. 41,191
Eric B. Meyertons	Reg. No. 34,876
Louise K. Miller	Reg. No. 36,609
David W. Quimby	Reg. No. 39,338
Larry D. Thompson	Reg. No. 43,952
David A. Rose	Reg. No. 26,223

each of said attorneys or agents being a member or an associate of the firm of Conley, Rose & Tayon, P.C., as attorney or agent for so long as they remain with such company or firm, with full power of substitution and revocation, to prosecute the application, to make alterations and amendments therein, to transact all business in the Patent and Trademark Office in connection therewith, and to receive the Letters Patent.

Please direct all communications to:

Robert C. Kowert
Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, Texas 78767-0398
Phone: (512) 476-1400

I hereby declare that all statements made herein of my own knowledge are true and that all statements made herein on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Inventor's Full Name: Kenneth A. Ward

Inventor's Signature: *Kenneth A. Ward* Date: 4/20/00

City and State (or Foreign Country) of Residence: Shrewsbury, MA Citizenship: U.S.A.

Post Office and Residence Address: 5 Joyce Circle, Shrewsbury, MA 01545
(Include number, street name, city, state and zip code)